In the Claims:

Please cancel claim 1 and amend claims 3, 4, 6, 11, 12, 13, 14, 15, 27, and 30 as follows:

1. (Cancelled)

2. (Cancelled)

3. (Currently Amended) A processor according to claim ~~2~~31, wherein the processor is arranged to preserve the second processor context, or that part of the second processor context which is different from the first processor context, when said exception has been generated.

4. (Currently Amended) A processor according to claim ~~2~~31, wherein the processor is arranged such that, when the processor is switched to a thread which is in the first execution mode, or when the processor is switched to a thread which was the last thread to be in the second execution mode, only the first processor context is preserved.

5. (Previously Presented) A processor according to claim 4, wherein the second processor context, or that part of the second processor context which is different from the first processor context, is preserved when the processor next enters the second execution mode to execute a thread other than a last thread to be in the second execution mode.

6. (Currently Amended) A processor according to claim ~~2~~31, wherein the number of threads that may be in the second execution mode at any one time is less than the total number of threads that may be active on the processor at any one time.

7. (Original) A processor according to claim 6, wherein the processor is arranged such that, when said exception has been generated, a check is carried out to determine whether the thread that caused the exception is allowed to enter the second execution mode.

8. (Previously Presented) A processor according to claim 7, wherein the check comprises determining whether that thread is a thread which is barred from the second execution mode.

9. (Original) A processor according to claim 7, wherein the check comprises determining whether a predetermined number of other threads are already in the second execution mode.

10. (Cancelled)

11. (Currently Amended) A processor according to claim ~~1~~31, wherein the processor is arranged to execute a first instruction set when in the first execution mode and a second instruction set when in the second execution mode.

12. (Currently Amended) A processor according to claim ~~1~~31, wherein the processor is switchable between a supervisor mode and a user mode, the user mode having restricted access to the processor's resources in comparison to the supervisor mode, and, when said exception is generated, the processor transfers from the user mode to the supervisor mode.

13. (Currently Amended) A processor according to claim ~~1~~31, the processor comprising at least one execution unit and a plurality of storage locations, the first processor context comprising the contents of storage locations accessible in the first execution mode and the second processor context comprising the contents of storage locations accessible in the second execution mode.

14. (Currently Amended) A processor according to claim ~~1~~31, the processor comprising a plurality of computational units for executing instructions in parallel, each computational unit having at least one execution unit and at least one storage location to which the execution unit has access.

15. (Currently Amended) A processor according to claim ~~1~~31, wherein the first execution mode is a scalar mode and the second execution mode is a parallel mode.

16-18. (Cancelled)

19. (Previously Presented) A processor according to claim 4, wherein the first execution mode is a scalar mode and the second execution mode is a parallel mode.

20-26. (Cancelled)

27. (Currently Amended) A method of operating a processor, the processor being switchable between a first execution mode and a second execution mode and having a first processor context when in the first execution mode and a second processor context, larger than the first processor context, when in the second execution mode, the method comprising executing a plurality of threads on a time share basis, the threads being able to change execution mode, and generating an exception when the processor attempts to change from one execution mode to the other to limit the number of threads that may be in the second execution mode at any one time, thereby limiting the number of times that the second processor context is preserved and restored~~.~~,

wherein the processor is arranged such that, if a predetermined number of other threads are already in the second execution mode, execution of the thread that caused the exception is suspended until the number of other threads that are in the second execution mode is less than the predetermined number.

28-29. (Cancelled)

30. (Currently Amended) A computer readable storage medium having stored thereon on operating system for a processor which is switchable between a first execution mode and a second execution mode and which has a first processor context when in the first execution mode and a second processor context, larger than the first processor context, when in the second execution mode, the operating system comprising a first program portion for switching execution between a plurality of threads on a time share basis, a second program portion allowing the threads to change execution mode, and exception handling program portion for handling an exception generated when the processor attempts to change from one execution mode to the other, and a third program portion for limiting the number of threads that may be in the second execution mode at any one time~~.~~,

wherein the processor is arranged such that, if a predetermined number of other threads are already in the second execution mode, execution of the thread that caused the exception is suspended until the number of other threads that are in the second execution mode is less than the predetermined number.

31. (Previously Presented) A processor switchable between a first execution mode and a second execution mode, the processor having a first processor context

when in the first execution mode and a second processor context, larger than the first processor context, when in the second execution mode, wherein the processor is arranged to execute a plurality of threads on a time share basis, the threads being able to change execution mode, and wherein the processor is arranged to generate an exception when the processor attempts to change from one execution mode to the other to keep track of when the execution modes are used and control which processor contexts are preserved at which times;

the number of threads in the second execution mode at any one time being limited, to limit the number of times that the second processor context is preserved and restored;

the number of threads that may be in the second execution mode at any one time being less than the total number of threads that may be active on the processor at any one time;

the processor being arranged such that, when said exception has been generated, a check is carried out to determine whether the thread that caused the exception is allowed to enter the second execution mode;

the check determining whether a predetermined number of other threads are already in the second execution mode;

wherein the processor is arranged such that, if a predetermined number of other threads are already in the second execution mode, execution of the thread that caused the exception is suspended until the number of other threads that are in the second execution mode is less than the predetermined number.